

Advanced Regular Expressions: Takeaways



by Dataquest Labs, Inc. - All rights reserved © 2019

Syntax

CAPTURE GROUPS

- Extracting text using a capture group:

```
s.str.extract(pattern_with_capture_group)
```

- Extracting text using multiple capture groups:

```
s.str.extract(pattern_with_multiple_capture_groups)
```

SUBSTITUTION

- Substituting a regex match:

```
s.str.replace(pattern, replacement_text)
```

Concepts

- Capture groups allow us to specify one or more groups within our match that we can access

Pattern	Explanation
<code>(yes)no</code>	Matches <code>yesno</code> , capturing <code>yes</code> in a single capture group.
<code>(yes)(no)</code>	Matches <code>yesno</code> , capturing <code>yes</code> and <code>no</code> in two capture groups. separately.

- Backreferences allow us to repeat a capture group within our regex pattern by referring to

Pattern	Explanation
<code>(yes)no\1</code>	Matches <code>yesnoyes</code>
<code>(yes)(no)\2\1</code>	Matches <code>yesnonoyes</code> them with an integer in the order they are captured.

- Lookarounds let us define a positive or negative match before or after our string.

Pattern	Explanation
<code>zzz(?=abc)</code>	Matches <code>zzz</code> only when it is followed by <code>abc</code>
<code>zzz(?!abc)</code>	Matches <code>zzz</code> only when it is not followed by <code>abc</code>
<code>(?<=abc)zzz</code>	Matches <code>zzz</code> only when it is preceded by <code>abc</code>
<code>(?<!zzz)abc</code>	Matches <code>abc</code> only when it is not preceded by <code>zzz</code>

Resources

- [re module](#)
- [RegExr Regular Expression Builder](#)



Takeaways by Dataquest Labs, Inc. - All rights reserved © 2019